psychotools - Infrastructure for Psychometric Modeling: Version 0.1-1

Zeileis, A ; Strobl, Carolin ; Wickelmaier, F

Abstract: Infrastructure for psychometric modeling such as data classes (e.g., for paired comparisons) and basic model fitting functions (e.g., for Rasch and Bradley-Terry models). Intended especially as a common building block for fitting psychometric mixture models in package ''psychomix'' and psychometric tree models in package ''psychotree''. License: GPL-2

Other titles: Package 'psychotools'

Package 'psychotools'

October 5, 2011

Title Infrastructure for Psychometric Modeling

Version 0.1-1

Date 2011-10-05

Authors@R c(person(given = ''Achim'', family = ''Zeileis'', role = c(''aut'', ''cre''), email = ''Achim.Zeileis@R-project.org''),person(given = ''Carolin'', family = ''Strobl'', role = ''aut'',email = ''Carolin.Strobl@psychologie.uzh.ch''), person(given = ''Florian'', family = ''Wickelmaier'', role = ''aut'', email = ''Florian.Wickelmaier@uni-tuebingen.de''))

Author Achim Zeileis [aut, cre], Carolin Strobl [aut], Florian Wickelmaier [aut]

Maintainer Achim Zeileis <Achim.Zeileis@R-project.org>

Depends R (>= 2.10.0), stats

Imports graphics, stats

Description Infrastructure for psychometric modeling such as data classes (e.g., for paired comparisons) and basic model fitting functions (e.g., for Rasch and Bradley-Terry models). Intended especially as a common building block for fitting psychometric mixture models in package ''psychomix'' and psychometric tree models in package ''psychotree''.

License GPL-2

Repository CRAN

Date/Publication 2011-10-05 11:07:31

R topics documented:

`btReg.fit` *Bradley-Terry Model Fitting Function*

Description

`btReg.fit` is a basic fitting function for simple Bradley-Terry models.

Usage

```
btReg.fit(y, weights = NULL, type = c("loglin", "logit"), ref = NULL,
  undecided = NULL, position = NULL, ...)
```

Arguments

`y`	paircomp object with the response.
`weights`	an optional vector of weights, interpreted as case weights (integer only).
`type`	character. Should an auxiliary log-linear Poisson model or logistic binomial be employed for estimation? The latter is only available if not undecided effects are estimated.
`ref`	character or numeric. Which object parameter should be the reference category, i.e., constrained to zero?
`undecided`	logical. Should an undecided parameter be estimated?
`position`	logical. Should a position effect be estimated?
`...`	further arguments passed to functions.

Details

`btReg.fit` provides a basic fitting function for simple Rasch models, intended as a building block for fitting Rasch trees and Rasch mixtures in the **psychotree** package, respectively.

`btReg.fit` returns an object of class `"btReg"` for which several basic methods are available, including `print`, `plot`, `summary`, `coef`, `vcov`, `logLik`, and `worth`.

Value

`btReg.fit` returns an S3 object of class `"btReg"`, i.e., a list with components as follows.

`coefficients`	estimated parameters on log-scale (without the first parameter which is always constrained to be 0),
`vcov`	covariance matrix of the parameters in the model,
`loglik`	log-likelihood of the fitted model,
`df`	number of estimated parameters,
`estfun`	empirical estimating function (also known as scores or gradient contributions),
`weights`	the weights used (if any),
`n`	number of observations (with non-zero weights),
`type`	character for model type (see above),
`ref`	character for reference category (see above),
`undecided`	logical for estimation of undecided parameter (see above),
`position`	logical for estimation of position effect (see above),
`labels`	character labels of the objects compared.

See Also

`RaschModel.fit`

Examples

```
## data
data("GermanParties2009", package = "psychotools")

## Bradley-Terry model
bt <- btReg.fit(GermanParties2009$preference)
summary(bt)
plot(bt)
```

`covariates` *Extract/Set Covariates*

Description

A generic function for extracting/setting covariates for an object.

Usage

```
covariates(object, ...)
covariates(object) <- value
```

Arguments

object	an object.
...	arguments passed to methods.
value	an object.

Examples

```
## method for "paircomp" data
pc <- paircomp(rbind(
  c(1,  1,  1), # a > b, a > c, b > c
  c(1,  1, -1), # a > b, a > c, b < c
  c(1, -1, -1), # a > b, a < c, b < c
  c(1,  1,  1)))
covariates(pc)
covariates(pc) <- data.frame(foo = factor(c(1, 2, 2), labels = c("foo", "bar")))
covariates(pc)
```

FirstNames *Popularity of First Names*

Description

Preferences of 192 respondents choosing among six boys names with respect to their popularity.

Usage

```
data("FirstNames")
```

Format

A data frame containing 192 observations on 11 variables.

preference Paired comparison of class paircomp. All 15 pairwise choices among six boys names: Tim, Lucas, Michael, Robin, Benedikt, and Julius.

ordered.pref Ordered paired comparison of class paircomp. Same as preference, but within-pair order is recognized.

gender Factor coding gender.

age Integer. Age of the respondents in years.

education Ordered factor. Level of education: 1 Hauptschule with degree (Secondary General School), 2 and 3 Realschule without and with degree (Intermediate Secondary School), 4 and 5 Gymnasium without and with degree (High School), 6 and 7 Studium without and with degree (University).

children Integer. Number of children.

state Factor. State of Germany where participant grew up.

state.reg Factor. The region (south, north-west, east) each state belongs to.

fname Factor. Participant's fist name(s). (Umlaute in Jörg and Jürgen have been transliterated to Joerg and Juergen for portability of the data.)

interviewer Factor. Interviewer id.

gender.int Factor coding interviewer's gender.

Details

A survey was conducted at the Department of Psychology, Universität Tübingen, in June 2009. The sample was stratified by gender and age (younger versus older than 30 years) with 48 participants in each group. The interviewers were Psychology Master's students who collected the data for course credits.

Participants were presented with 15 pairs of boys names in random order. On each trial, their task was to choose the name they would rather give to their own child. The pairs of boys names were read to the participants one at a time. A given participant compared each pair in one order only, hence the NA's in `ordered.pref`.

The names were selected to fall within the upper (Tim, Lucas), mid (Michael, Robin) and lower (Benedikt, Julius) range of the top 100 of the most popular boys names in Germany in the years from 1990 to 1999 (http://www.beliebte-vornamen.de/3778-1990er-jahre.htm). The names have either front (e, i) or back (o, u) vowels in the stressed syllables. Phonology of the name and attractiveness of a person have been shown to be related (Perfors, 2004; Hartung et al., 2009).

References

Hartung, F., Klenovsak, D., Santiago dos Santos, L., Strobl, C., and Zaefferer, D. (2009). Are Tims Hot and Toms not? Probing the effect of sound symbolism on perception of facial attractiveness. Presented at the *31th Annual Meeting of the Cognitive Science Society*, July 27 to August 1, Amsterdam, The Netherlands.

Perfors, A. (2004). What's in a Name? The effect of sound symbolism on perception of facial attractiveness. Presented at the *26th Annual Meeting of the Cognitive Science Society*, August 5-7, Chicago, USA.

See Also

`paircomp`

Examples

```
data("FirstNames", package = "psychotools")
summary(FirstNames$preference)
covariates(FirstNames$preference)
```

GermanParties2009 *Choice among German Political Parties*

Description

Preferences of 192 respondents choosing among five German political parties and abstention from voting.

Usage

```
data("GermanParties2009")
```

Format

A data frame containing 192 observations on 6 variables.

preference Paired comparison of class `paircomp`. All 15 pairwise choices among five German parties and abstention from voting.

ordered.pref Ordered paired comparison of class `paircomp`. Same as `preference`, but within-pair order is recognized.

gender Factor coding gender.

age Integer. Age of the respondents in years.

education Ordered factor. Level of education: 1 no degree, 2 Hauptschule (Secondary General School), 3 Realschule (Intermediate Secondary School), 4 Gymnasium (High School), 5 Studium (University)

crisis Factor. Do you feel affected by the economic crisis?

interviewer Factor. Interviewer id.

Details

A survey was conducted at the Department of Psychology, Universität Tübingen, in June 2009, three months before the German election. The sample was stratified by gender and age (younger versus older than 30 years) with 48 participants in each group.

The parties to be compared were Die Linke (socialists), Die Grünen (ecologists), SPD (social democrats), CDU/CSU (conservatives), and FDP (liberals). In addition, there was the option of abstaining from voting (coded as `none`).

Participants were presented with 15 pairs of options in random order. On each trial, their task was to choose the party they would rather vote for at an election for the German parliament. A given participant compared each pair in one order only, hence the NA's in `ordered.pref`.

In order to minimize response biases, the pairs of options were read to the participants one at a time. Participants made their choices by crossing either "First Option" or "Second Option" on an anonymous response sheet.

The interviewers were Psychology Master's students who collected the data for course credits. Since they mainly interviewed people they knew, the results are not representative of the political opinions

in Germany. As far as the winner of the survey (Die Grünen) is concerned, however, the results agree with the outcome of the election for the Tübingen voters.

The results of the election on September 27, 2009 (number of so-called Zweitstimmen in percent) were:

	Germany	Tübingen
Die Linke	11.9	8.5
Die Grünen	10.7	27.9
SPD	23.0	21.1
CDU/CSU	33.8	23.0
FDP	14.6	13.9
Others	6.0	5.7

The voter turnout was 70.8 percent in Germany and 80.5 percent in Tübingen.

See Also

pairscomp

Examples

```
data("GermanParties2009", package = "psychotools")
summary(GermanParties2009$preference)
```

---

labels<- *Set Labels*

Description

A generic function for setting labels for an object.

Usage

```
labels(object) <- value
```

Arguments

object	an object.
value	an object.

Examples

```
## method for "paircomp" data
pc <- paircomp(rbind(
  c(1,  1,  1), # a > b, a > c, b > c
  c(1,  1, -1), # a > b, a > c, b < c
  c(1, -1, -1), # a > b, a < c, b < c
  c(1,  1,  1)))
labels(pc)
labels(pc) <- c("ah", "be", "ce")
pc
```

---

mscale *Extract/Replace Measurement Scale*

Description

Generic functions for extracting and replacing the measurement scale from an object.

Usage

```
mscale(object, ...)
mscale(object) <- value
```

Arguments

object	an object.
...	arguments passed to methods.
value	an object describing the measurement scale.

Examples

```
## methods for "paircomp" data
pc <- paircomp(rbind(
  c(2,  1,  0),
  c(1,  1, -1),
  c(1, -2, -1),
  c(0,  0,  0)))
pc

## extract
mscale(pc)

## replace (collapse to >/=/< scale)
mscale(pc) <- sign(mscale(pc))
pc
```

paircomp *Data Structure for Paired Comparisons*

Description

A class for representing data from paired comparison experiments along with methods for many generic functions.

Usage

```
paircomp(data,
  labels = NULL, mscale = NULL, ordered = FALSE, covariates = NULL)
```

Arguments

data	matrix. A matrix with integer values where the rows correspond to subjects and the columns to paired comparisons between objects. See below for details.
labels	character. A vector of character labels for the objects. By default a suitable number of letters is used.
mscale	integer. A vector of integers giving the measurement scale. See below for details. By default guessed from data.
ordered	logical. Does data contain both orderings of each comparison?
covariates	data.frame. An optional data.frame with object covariates, i.e., it must have the same number of rows as the length of labels. May be NULL (default).

Details

paircomp is designed for holding paired comparisons of k objects measured for n subjects.

The comparisons should be coded in an integer matrix data with n rows (subjects) and $\binom{k}{2}$ columns (unless ordered = TRUE, see below). The columns must be ordered so that objects are sequentially compared with all previous objects, i.e.: 1:2, 1:3, 2:3, 1:4, 2:4, 3:4, etc. Each column represents the results of a comparison for two particular objects. Positive values signal that the first object was preferred, negative values that the second was preferred, zero signals no preference. Larger absolute values signal stronger preference.

mscale provides the underlying measurement scale. It must be a symmetric sequence of integers of type (-i):i where i must be at least 1. However, it may exclude 0 (i.e., forced choice).

If ordered = TRUE, the order of comparison matters and thus data is assumed to have twice as many columns. The second half of columns then corresponds to the comparisons 2:1, 3:1, 3:2, 4:1, 4:2, 4:3, etc.

Value

paircomp returns an object of class "paircomp" which is a matrix (essentially data) with all remaining arguments of paircomp as attributes (after being checked and potentially suitably coerced or transformed).

See Also

subset.paircomp, print.paircomp

Examples

```
## a simple paired comparison
pc <- paircomp(rbind(
  c(1,  1,  1), # a > b, a > c, b > c
  c(1,  1, -1), # a > b, a > c, b < c
  c(1, -1, -1), # a > b, a < c, b < c
  c(1,  1,  1)))

## basic methods
pc
str(pc)
summary(pc)
pc[2:3]
c(pc[2], pc[c(1, 4)])

## methods to extract/set attributes
labels(pc)
labels(pc) <- c("ah", "be", "ce")
pc
mscale(pc)
covariates(pc)
covariates(pc) <- data.frame(foo = factor(c(1, 2, 2), labels = c("foo", "bar")))
covariates(pc)
names(pc)
names(pc) <- LETTERS[1:4]
pc

## reorder() and subset() both select a subset of
## objects and/or reorders the objects
reorder(pc, c("ce", "ah"))


## include paircomp object in a data.frame
## (i.e., with subject covariates)
dat <- data.frame(
  x = rnorm(4),
  y = factor(c(1, 2, 1, 1), labels = c("hansi", "beppi")))
dat$pc <- pc
dat


## formatting with long(er) labels and extended scale
pc2 <- paircomp(rbind(
  c(4,  1,  0),
  c(1,  2, -1),
  c(1, -2, -1),
  c(0,  0,  -3)),
  labels = c("Nordrhein-Westfalen", "Schleswig-Holstein", "Baden-Wuerttemberg"))
```

```
## default: abbreviate
print(pc2)
print(pc2, abbreviate = FALSE)
print(pc2, abbreviate = FALSE, width = FALSE)


## paired comparisons with object covariates
pc3 <- paircomp(rbind(
  c(2,  1,  0),
  c(1,  1, -1),
  c(1, -2, -1),
  c(0,  0,  0)),
  labels = c("New York", "Rio", "Tokyo"),
  covariates = data.frame(hemisphere = factor(c(1, 2, 1), labels = c("North", "South"))))
covariates(pc3)
```

---

plot.btReg *Visualizing Simple Rasch Models*

Description

Base graphics plotting function for Rasch models.

Usage

```
## S3 method for class 'btReg'
plot(x, worth = TRUE, index = TRUE, names = TRUE,
  ref = TRUE, abbreviate = FALSE, type = NULL, lty = NULL,
  xlab = "Objects", ylab = NULL, ...)
```

Arguments

x	an object of class "btReg".
worth	logical. Should worth parameters (or alternatively coefficients on log-scale) be displayed?
index	logical. Should different indexes for different items be used?
names	logical. Should the names for the objects be displayed?
ref	logical. Should a horizontal line for the reference level be drawn?
abbreviate	logical or numeric. Should object names be abbreviated? If numeric this controls the length of the abbreviation.
type	plot type. Default is "b" if index is TRUE.
lty	line type.
xlab, ylab	x and y axis labels.
...	further arguments passed to plot.

See Also

btReg.fit

Examples

```
## data
data("GermanParties2009", package = "psychotools")

## Bradley-Terry model
bt <- btReg.fit(GermanParties2009$preference)
plot(bt)
plot(bt, worth = FALSE)
plot(bt, index = FALSE)
```

---

plot.paircomp *Plotting Paired Comparison Data*

Description

Plotting the frequency table from "paircomp" data.

Usage

```
  ## S3 method for class 'paircomp'
plot(x, off = 0.05,
  xlab = "Proportion of comparisons", ylab = "", tol.xlab = 0.05,
  abbreviate = TRUE, hue = NULL, chroma = 40, luminance = 80,
  xlim = c(0, 1), ylim = NULL, xaxs = "i", yaxs = "i", ...)
```

Arguments

x	an object of class "paircomp".
off	numeric. Offset between segments on the y-axis.
xlab, ylab	character. Axis labels.
tol.xlab	numeric. convenience tolerance parameter for x-axis annotation. If the distance between two labels drops under this threshold, they are plotted equidistantly.
abbreviate	logical or integer. Should object labels be abbreviated? Alternative an integer with the desired abbreviation length. The default is some heuristic based on the length of the labels.
hue	numeric. A vector of hues in [0, 360], recycled to the number of objects compared in x. A sequential palette is computed for each hue, see below.
chroma	numeric. Maximum chroma in the palette.
luminance	numeric. Minimum (and maximum) luminance in the palette. If omitted, the maximum is set to 95.
xlim, ylim, xaxs, yaxs, ...	graphical arguments passed to plot.

Details

The `plot` method creates a frequency table (using `summary`) and visualizes this using a sort of spine plot with HCL-based diverging palettes. See Zeileis, Hornik, Murrell (2009) for the underlying ideas.

References

Zeileis A., Hornik K. and Murrell P. (2009), Escaping RGBland: Selecting Colors for Statistical Graphics. *Computational Statistics & Data Analysis*, **53**, 3259-3270. doi:10.1016/j.csda.2008.11.033 Preprint available from http://statmath.wu.ac.at/~zeileis/papers/Zeileis+Hornik+Murrell-2009.pdf.

See Also

`paircomp`

Examples

```
data("GermanParties2009", package = "psychotools")
par(mar = c(5, 6, 3, 6))
plot(GermanParties2009$preference, abbreviate = FALSE)
```

---

plot.RaschModel *Visualizing Simple Rasch Models*

Description

Base graphics plotting function for Rasch models.

Usage

```
## S3 method for class 'RaschModel'
plot(x, difficulty = TRUE,
  center = TRUE, index = TRUE, names = NULL, abbreviate = FALSE, ref = TRUE,
 col = cbind("lightgray", "black"), refcol = "lightgray", linecol = "black", lty = 2,
 cex = 1, pch = cbind(19, 1), type = NULL, ylim = NULL, xlab = "Items", ylab = NULL, ...)
```

Arguments

`x`	an object of class `"RaschModel"`.
`difficulty`	logical. Should item difficulty (or alternatively: easiness) parameters be displayed?
`center`	logical. Should the item parameters be centered?
`index`	logical. Should different indexes for different items be used?
`names`	logical. Should the names for the objects be displayed?

abbreviate	logical or numeric. Should object names be abbreviated? If numeric this controls the length of the abbreviation.
ref	logical. Should a horizontal line for the reference level be drawn?
col, pch, cex	graphical appearance of plotting symbols. Can be of the same length as number of items. Additionally col and pch can be matrices with two columns resulting in two symbols being overplotted.
refcol	line color for reference line (if ref).
linecol	line color.
lty	line type.
type	plot type. Default is "b" if index is TRUE.
ylim	y axis limits.
xlab, ylab	x and y axis labels.
...	further arguments passed to plot.

See Also

RaschModel.fit

---

print.paircomp *Formatting Paired Comparison Data*

Description

Fine control for formatting and printing objects of "paircomp" data.

Usage

```
  ## S3 method for class 'paircomp'
format(x, sep = ", ", brackets = TRUE,
    abbreviate = NULL, width = getOption("width") - 7, ...)
  ## S3 method for class 'paircomp'
print(x, quote = FALSE, ...)
```

Arguments

x	an object of class "paircomp".
sep	character. A character for separating comparisons within subjects.
brackets	logical or character. Either a logical (Should brackets be wrapped around all comparisons for a single subject?) or a character of length two with opening and ending symbol.
abbreviate	logical or integer. Should object labels be abbreviated? Alternative an integer with the desired abbreviation length. The default is some heuristic based on the length of the labels.

width	integer or logical. Maximal width of the string for a subject. If FALSE no maximal width is set.
...	arguments passed to other functions.
quote	logical. Should quotes be printed?

Details

The print method just calls format (passing on all further arguments) and then prints the resulting string.

See Also

paircomp

Examples

```
pc2 <- paircomp(rbind(
  c(4,  1,  0),
  c(1,  2, -1),
  c(1, -2, -1),
  c(0,  0,  -3)),
  labels = c("New York", "Rio", "Tokyo"))

print(pc2)
print(pc2, abbreviate = FALSE)
print(pc2, abbreviate = FALSE, width = 10)
```

RaschModel.fit *Rasch Model Fitting Function*

Description

RaschModel.fit is a basic fitting function for simple Rasch models.

Usage

```
RaschModel.fit(y, weights = NULL, start = NULL, gradtol = 1e-6,
  deriv = c("sum", "diff", "numeric"), hessian = TRUE, ...)
```

Arguments

y	object that can be coerced (via as.matrix) to a binary 0/1 matrix.
weights	an optional vector of weights, interpreted as case weights (integer only).
start	an optional vector of starting values.
deriv	character. Which type of derivatives should be used for computing gradient and Hessian matrix? Analytical with sum algorithm ("sum"), analytical with difference algorithm ("diff", faster but numerically unstable), or numerical.

hessian	logical. Should the Hessian of the final model be computed? If set to FALSE, the vcov method can only return NAs and consequently no standard errors or tests are available in the summary.
gradtol, ...	further arguments passed to nlm.

Details

RaschModel.fit provides a basic fitting function for simple Rasch models, intended as a building block for fitting Rasch trees and Rasch mixtures in the **psychotree** package, respectively.

RaschModel.fit returns an object of class "RaschModel" for which several basic methods are available, including print, plot, summary, coef, vcov, logLik, and worth.

Value

RaschModel.fit returns an S3 object of class "RaschModel", i.e., a list with components as follows.

coefficients	estimated item difficulty parameters (without first item parameter which is always constrained to be 0),
vcov	covariance matrix of the parameters in the model,
loglik	log-likelihood of the fitted model,
df	number of estimated parameters,
data	the original data supplied (excluding columns without variance),
weights	the weights used (if any),
n	number of observations (with non-zero weights),
items	status indicator (0, 0/1, 1) of all original items,
na	logical indicating whether the data contains NAs,
elementary_symmetric_functions	List of elementary symmetric functions for estimated parameters (up to order 2; or 1 in case of numeric derivatives),
nlm_code	convergence code from nlm,
iterations	number of iterations used by nlm,
gradtol	tolerance passed to nlm.

See Also

btReg.fit

Examples

```
## Verbal aggression data
data("VerbalAggression", package = "psychotools")

## Rasch model for the self-to-blame situations
m <- RaschModel.fit(VerbalAggression$resp2[, 1:12])
summary(m)
plot(m)
```

SoundQuality *Quality of Multichannel Reproduced Sound*

Description

Paired comparison judgments of 40 selected listeners with respect to eight audio reproduction modes and for types of music.

Usage

```
data("SoundQuality")
```

Format

A data frame containing 783 observations on 6 variables.

id Factor. Listener ID.

time Factor. Listening experiment before or after elicitation and scaling of more specific auditory attributes.

progmat Factor. The Program material: Beethoven, Rachmaninov, Steely Dan, Sting.

repet The repetition within each time point.

session The experimental session coding the presentation order of the program material.

preference Paired comparison of class `paircomp`. Preferences for all 28 paired comparisons from 8 audio reproduction modes: Mono, Phantom Mono, Stereo, Wide-Angle Stereo, 4-channel Matrix, 5-channel Upmix 1, 5-channel Upmix 2, and 5-channel Original.

Details

The data were collected within a series of experiments conducted at the Sound Quality Research Unit (SQRU), Department of Acoustics, Aalborg University, Denmark, between September 2004 and March 2005.

The results of scaling listener preference and spatial and timbral auditory attributes are reported in Choisel and Wickelmaier (2007).

Details about the loudspeaker setup and calibration are given in Choisel and Wickelmaier (2006).

The attribute elicitation procedure is described in Wickelmaier and Ellermeier (2007) and in Choisel and Wickelmaier (2006).

The selection of listeners for the experiments is described in Wickelmaier and Choisel (2005).

References

Choisel, S. & Wickelmaier, F. (2006). Extraction of auditory features and elicitation of attributes for the assessment of multichannel reproduced sound. Journal of the Audio Engineering Society, 54(9), 815-826.

Choisel, S. & Wickelmaier, F. (2007). Evaluation of multichannel reproduced sound: Scaling auditory attributes underlying listener preference. Journal of the Acoustical Society of America, 121(1), 388-400.

Wickelmaier, F. & Choisel, S. (2005). Selecting participants for listening tests of multichannel reproduced sound. Presented at the AES 118th Convention, May 28-31, Barcelona, Spain, convention paper 6483.

Wickelmaier, F. & Ellermeier, W. (2007). Deriving auditory features from triadic comparisons. Perception & Psychophysics, 69(2), 287-297.

See Also

paircomp

Examples

```
data("SoundQuality", package = "psychotools")
summary(SoundQuality$preference)
ftable(xtabs(~ time + repet + progmat, data = SoundQuality))
```

---

subset.paircomp *Subsetting/Reordering Paired Comparison Data*

Description

Selection of subsets of objects to be compared and/or reordering of objects in "paircomp" data.

Usage

```
  ## S3 method for class 'paircomp'
reorder(x, labels, ...)
  ## S3 method for class 'paircomp'
subset(x, subset, select, ...)
```

Arguments

x	an object of class "paircomp".
labels, select	character or integer. Either a vector of (at least two) elements of labels(x) or an integer with their position. Partial string matching is enabled.
subset	currently not implemented. (Should be a specification of subsets of subjects.)
...	currently not used.

Details

The subset method currently just calls the reorder method.

See Also

paircomp

Examples

```
pc <- paircomp(rbind(
  c(1,  1,  1), # a > b, a > c, b > c
  c(1,  1, -1), # a > b, a > c, b < c
  c(1, -1, -1), # a > b, a < c, b < c
  c(1,  1,  1)))
reorder(pc, c("c", "a"))
```

VerbalAggression *Situation-Response Questionnaire on Verbal Aggression*

Description

Responses of 316 subjects to 24 items describing possible reactions to 4 different frustrating situations.

Usage

```
data("VerbalAggression")
```

Format

A data frame containing 316 observations on 4 variables.

resp Item response matrix with values 0/1/2 coding no/perhaps/yes, respectively.

resp2 Dichotomized item response matrix with perhaps/yes merged to 1.

gender Factor coding gender.

anger Trait anger, assessed by the Dutch adaptation of the state-trait anger scale (STAS).

Details

The 24 items are constructed by factorial combination of four different frustrating situations (see below), three possible verbally aggressive responses (curse, scold, shout), and two behavioural models (want, do). The four situations are

S1: A bus fails to stop for me.
S2: I miss a train because a clerk gave me faulty information.
S3: The grocery store closes just as I am about to enter.
S4: The operator disconnects me when I used up my last 10 cents for a call.

Note that the first two situations are other-to-blame situations, and the latter two are self-to-blame situations.

The subjects were 316 first-year psychology students from a university in the Dutch speaking part of Belgium. Participation was a partial fulfillment of the requirement to participate in research. The sample consists of 73 males and 243 females, reflecting the gender proportion among psychology students. The average age was 18.4.

Source

Onine materials accompanying De Boeck and Wilson (2004).

http://bear.soe.berkeley.edu/EIRM/

References

De Boeck, P., Wilson, M. (eds) (2004). Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach. New York: Springer-Verlag.

Smits, D.J.M., De Boeck, P., Vansteelandt, K. (2004). The Inhibition of Verbally Aggressive Behaviour *European Journal of Personality*, **18**, 537-555. doi:10.1002/per.529

See Also

RaschModel.fit

Examples

```
data("VerbalAggression", package = "psychotools")

## Rasch model for the self-to-blame situations
m <- RaschModel.fit(VerbalAggression$resp2[, 1:12])
summary(m)
plot(m)
```

---

worth *Extract Worth Parameters*

Description

Generic functions for extracting worth parameters from paired comparison models.

Usage

```
worth(object, ...)
```

Arguments

object	an object.
...	arguments passed to methods.

See Also

btReg.fit, RaschModel.fit

Examples

```
## data
data("GermanParties2009", package = "psychotools")

## Bradley-Terry model
bt <- btReg.fit(GermanParties2009$preference)

## worth parameters
worth(bt)
```

Index